WEST Search History

DATE: Wednesday, December 03, 2003

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT,PGPB; PLUR=YES; OP=ADJ		
L20	L16 and trigger	19	L20
L19	L16 and temporal	4	L19
L18	L16 and (e language)	0	L18
L17	L16 and (underlying control)	0	L17
L16	L15 and (dynamic or static)	47	L16
L15	L14 and l5	74	L15
L14	L11 and (parsing or parse or parsed)	159	L14
L13	L12 and l5	46	L13
L12	L11 and (scan or scanned or scanning)	133	L12
L11	L10 and (data flow or control structure or node)	260	L11
L10	L8	352	L10
	DB=JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L9	L8	7	L9
	DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L8	L7 and ((map$ or translat$) near3 (hardware or HDL))	359	L8
L7	L6 and (synthesi$ or verification)	5185	L7
L6	hardware description or HDL	9336	L6
	DB=USPT,PGPB; PLUR=YES; OP=ADJ		
L5	L4 or l3 or l2 or l1	8271	L5
L4	((716/2 |716/3 |716/4 |716/5)!.CCLS.)	2326	L4
L3	((704/220)!.CCLS.)	287	L3
L2	((703/1 |703/2 |703/3 |703/4 |703/13 |703/14 |703/15 |703/22 |703/26 |703/27 |703/28)!.CCLS.)	3593	L2
L1	((717/106 |717/107 |717/108 |717/109 |717/119 |717/134 |717/135 |717/136 |717/137 |717/138 |717/139 |717/140 |717/141 |717/142 |717/143 |717/144 |717/145 |717/146 |717/147 |717/148 |717/149 |717/150 |717/151 |717/152 |717/153 |717/154 |717/155 |717/156 |717/157)!.CCLS.)	2474	L1

END OF SEARCH HISTORY

Search Results - Record(s) 1 through 7 of 7 returned.

☐ 1. Document ID: US 6601024 B1

L9: Entry 1 of 7 File: DWPI Jul 29, 2003

DERWENT-ACC-NO: 2003-707679
DERWENT-WEEK: 200367

TITLE: Hardware description language based design translation method in computer aided design, involves creating secondary module set using RTL format conversion and integrating modules to obtain RTL and gate level combined design

INVENTOR: BHATT, M; CHONNAD, S S ; KANDASWAMY, M E ; KOPETZKY, C A ; SAVAGE, T W

PRIORITY-DATA: 1998US-0191153 (November 12, 1998)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
US 6601024 B1	July 29, 2003		021	G06F017/50

INT-CL (IPC): G06 F 17/50

ABSTRACTED-PUB-NO: US 6601024B
BASIC-ABSTRACT:

NOVELTY - A reference gate level net list (255) is created by synthesizing a primary register transfer level (RTL) format. A secondary set of modules (222, 250,252, 254) is created by translating the RTL format of the primary set of modules to a secondary RTL format. A combined RTL and gate level design is created by integrating specific modules from the secondary set of modules within the net list.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for translation verification method.

USE - For translating hardware description language (HDL) based design specifications during computer aided designing of circuits like very high speed integrated circuit (VHSIC), application specific integrated circuit (ASIC) and field programmable gate array (FPGA).

ADVANTAGE - Reduces total translation time due to use of RTL formats and ensures exact functional equivalence between the initial and target RTL based designs.

DESCRIPTION OF DRAWING(S) - The figure shows the partial functional block diagram of the verification system.

modules 222,250,252,254

net list 255

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 2. Document ID: US 6564354 B1

L9: Entry 2 of 7 File: DWPI May 13, 2003

DERWENT-ACC-NO: 2003-554584
DERWENT-WEEK: 200352

TITLE: Conditional expressions translation method for hardware logic design testing, involves synthesizing always statement assigned with global variable to generate logic circuit representing non-Verilog HDL program

INVENTOR: HYLANDER, P D; WANG, L

PRIORITY-DATA: 2000US-0586167 (June 1, 2000)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
US 6564354 B1	May 13, 2003		025	G06F017/50

INT-CL (IPC): G06 F 17/50

ABSTRACTED-PUB-NO: US 6564354B
BASIC-ABSTRACT:

NOVELTY - A global variable representing a variable of IF/CASE/COND (ICC) expressions is created and assigned to a variable placed in an always statement recognized by a hardware description language (HDL) program, when ICC expressions occurring within the program are part of a conditional/binary operator expression. The always statement is synthesized to generate a logic circuit representing a non-Verilog HDL program.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for computer readable medium storing conditional expressions translation program.

USE - For translating condition expressions of non-Verilog hardware description language (HDL) program including simple ICC expressions, nested ICC expressions, special expressions and edge triggered statements, to Verilog HDL format used for logic design of hardware such as complex CPU.

ADVANTAGE - Accurate translates all ICC conditional expressions occurring within non-Verilog HDL, including nested ICC expressions, to Verilog HDL.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart illustrating the translation method of non-Verilog HDL conditional expression occurring in a non-Verilog HDL program.

☐ 3. Document ID: US 6167363 A

L9: Entry 3 of 7 File: DWPI Dec 26, 2000

DERWENT-ACC-NO: 2001-225705
DERWENT-WEEK: 200123

TITLE: Comparison of model simulation output signal of logic circuit, involves starting RTL and HDL model simulation in two processes, respectively of operating system supporting interprocess communication to run processes

INVENTOR: STAPLETON, W G

PRIORITY-DATA: 1995US-005045P (October 10, 1995), 1996US-0592041 (January 26, 1996), 1998US-0052897 (March 31, 1998)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
US 6167363 A	December 26, 2000		014	G06F017/50

INT-CL (IPC): G06 F 17/50

ABSTRACTED-PUB-NO: US 6167363A
BASIC-ABSTRACT:

NOVELTY - The RTL and HDL model simulations of the logic circuit is started in first and second processes, respectively of an operating system supporting an interprocess communication mechanism. At the end of a predetermined simulation period, the values of the output signals of the RTL model are compared with the values of the output signals of the HDL model to check whether the values are equal or not.

DETAILED DESCRIPTION - The method involves running the RTL model simulation in a first process of an operating system for a predetermined simulation time period using an input test vector. At the end of the time period, values of output signals of RTL model are retrieved. The output values are sent to the second process of the operating system, through the interprocess communication mechanism. The HDL model simulation is run for the predetermined simulation period using the input test vector. At the end of the time period, values of the output signals of the HDL model are retrieved. Output values of RTL model and HDL model are compared to check whether they are equal or not. The procedure is repeated until a predetermined number of simulation cycles are completed.

USE - For comparison of output signals of register transfer level (RTL) and hardware description language (HDL) model simulation of logic such as microprocessor circuits in computer aided design tools for simulation or verification.

ADVANTAGE - Provides a simulator that is tailored for use in the design of complex logic circuit such as microprocessor. The simulator is fast and cycle accurate. Translation from the RTL objects to the HDL represented logic circuit is efficiently performed.

DESCRIPTION OF DRAWING(S) - The figure shows the flow diagram of communication between RTL model and control program which monitors the simulation of the HDL model.

☐ 4. Document ID: US 6026219 A

L9: Entry 4 of 7 File: DWPI Feb 15, 2000

DERWENT-ACC-NO: 2000-222680
DERWENT-WEEK: 200339

TITLE: Data processing method for coupling results of behavioral synthesis with those of logic synthesis, by analyzing digital circuit using HDL source description from which digital circuit was created

INVENTOR: KNAPP, D W; LY, T A ; MACMILLEN, D B ; MILLER, R A

PRIORITY-DATA: 1995US-0440101 (May 12, 1995)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
US 6026219 A	February 15, 2000		034	G06F019/00

INT-CL (IPC): G06 F 19/00

ABSTRACTED-PUB-NO: US 6026219A
BASIC-ABSTRACT:

NOVELTY - The method for coupling the results of behavioral synthesis with logic synthesis involves using a timing verifier to pre-calculate the timing characteristics of a circuit for use by behavioral synthesis. Timing for control chaining is included in the pre-calculated timing characteristics. Once behavioral synthesis is complete, logic synthesis is informed of timing constraints introduced by behavioral synthesis.

DETAILED DESCRIPTION - The method involves computing ready times of operations in a circuit using a timing verifier. The circuit is mapped to a target technology and a conventional timing verifier is used to time the circuit. The ready times of the operators in the circuit are stored in chaining tables. The delay information in the chaining tables can be used by scheduling to determine which operations can be chained into a single cycle.

An INDEPENDENT CLAIM is included for a system for building a digital circuit representation.

USE - Linking results from a logic synthesis system to behavioral synthesis system in computer aided design for digital circuits.

ADVANTAGE - Provides method for post annotating multi-cycle constraints. Annotations allow the logic optimizer to correctly optimize multi-cycle operations.

DESCRIPTION OF DRAWING(S) - The drawing shows a block diagram showing the components of a synthesis system of the invention.

Behavioral synthesis 1300

HDL translator 1310

Technology library 1320

Logic synthesis 1330

Timing verifier 1340

☐ 5. Document ID: JP 10254915 A JP 3162316 B2

L9: Entry 5 of 7 File: DWPI Sep 25, 1998

DERWENT-ACC-NO: 1998-573254
DERWENT-WEEK: 200126

TITLE: Logic circuit tester for e.g. in-circuit tester, LSI tester - includes programmable device connected to tested circuit, and control unit controlling writing-in of mapping data to programmable device

PRIORITY-DATA: 1997JP-0054432 (March 10, 1997)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 10254915 A	September 25, 1998		013	G06F017/50
JP 3162316 B2	April 25, 2001		012	G01R031/28

INT-CL (IPC): G01 R 31/28; G06 F 17/50

ABSTRACTED-PUB-NO: JP 10254915A
BASIC-ABSTRACT:

The apparatus includes a data conversion unit (3) and a logic synthesis circuit (8). The logic synthesis circuit produces a network list functionally equivalent to the

process result of the conversion unit. Based on the network list, a technology mapping unit (10) produces mapping data. A programmable device (13) is operated by a basic timing signal and assembles circuit equivalent to HDL data based on the mapping data.

A control unit (19) controls the writing-in operation of the mapping data to the device, and in generation of basic timing signal. The device is connected to the logic circuit. The electrical property, logical correctness, and defects in the circuit are measured.

ADVANTAGE - Needs reduced memory capacity. Attains environment equivalent to HDL test bench.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Clip Img | Image

---

☐ 6. Document ID: JP 09282336 A

L9: Entry 6 of 7 File: DWPI Oct 31, 1997

DERWENT-ACC-NO: 1998-024049
DERWENT-WEEK: 199803

TITLE: Automatic synthesis system of logic circuit using CAD - has structural transition part which freely converts logic description containing description of part mapping designation in in-house data structure

PRIORITY-DATA: 1996JP-0115738 (April 12, 1996)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 09282336 A	October 31, 1997		005	G06F017/50

INT-CL (IPC): G06 F 17/50

ABSTRACTED-PUB-NO: JP 09282336A
BASIC-ABSTRACT:

The system has a logic description (1) containing the description of part mapping designation in the in-house data structural transition part (2). One part of logic description described by HDL is subject to part mapping and is converted to an in-house data structure (3). A block information in the description of part mapping designation is formed with the reference of a library.

A part mapping unit (4) designates that the part which is subject to part mapping is not replaced again with another function block and designates only the part into which is not done. A logic synthesis unit (6) performs a logic performs a logic synthesis, thereby, synthesising the circuit.

ADVANTAGE - Enables simple and easy operation.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Clip Img | Image

---

☐ 7. Document ID: WO 9602038 A1 US 5519627 A US 5491640 A

L9: Entry 7 of 7 File: DWPI Jan 25, 1996

DERWENT-ACC-NO: 1996-097738
DERWENT-WEEK: 199626

TITLE: Integrated circuit fabrication method for synthesising datapaths in IC design

- using datapath synthesiser which maps HDL specification to both combinational and sequential components available in datapath library

INVENTOR: MAHMOOD, M; SHARMA, B K ; KINGSLEY, C H

PRIORITY-DATA: 1994US-0272205 (July 8, 1994), 1992US-0877951 (May 1, 1992), 1994US-0339928 (November 15, 1994)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
WO 9602038 A1	January 25, 1996	E	040	G06F017/50
US 5519627 A	May 21, 1996		028	G06F017/50
US 5491640 A	February 13, 1996		020	G06F015/60

INT-CL (IPC): G06 F 15/60; G06 F 17/50

ABSTRACTED-PUB-NO: US 5491640A
BASIC-ABSTRACT:

The method of fabricating an integrated circuit involves developing a set of circuit specifications for an integrated circuit and encoding the set of specifications into a hardware description language and inputting the hardware description language into a computer. A datapath library is provided which includes both sequential and combinational components. A netlist including a sequential component derived from the datapath library, is synthesised from the hardware description language, and an integrated circuit is fabricated as specified by the netlist. The netlist further includes a combinational component and random logic.

The synthesising involves parsing the HDL into a controlled data flow graph (CDFG), generating an IC expression tree for each output of the IC in terms of primary inputs and constraints using the CDFGs, and partitioning each IC expression tree into random logic and datapath categories. The random logic is synthesised using random logic synthesis to create a random logic netlist, and the datapath is synthesised using datapath synthesis to create a datapath netlist including both sequential and combinational components. The datapath and random logic netlists are combined to develop an IC netlist.

USE/ADVANTAGE - Synthesising datapaths for IC design and fabrication. Enables sequential logic to be synthesised in datapath synthesizer using optimised library of sequential components and optimisation criteria specified by circuit designer.
ABSTRACTED-PUB-NO:

US 5519627A EQUIVALENT-ABSTRACTS:

A method for fabricating an integrated circuit comprising:

developing a set of circuit specifications for an integrated circuit;

encoding said set of circuit specifications in a hardware description language and inputting said hardware description language into a digital computer;

providing a datapath library including both sequential components and combinational components, wherein said sequential components exhibit sequential logic having an output dependent on inputs and also dependent on a state of a memory unit included within said sequential component, and wherein said combinational components exhibit combinational logic having an output only dependent on inputs to said combinational logic;

synthesizing from said hardware description language and on said computer a netlist including a sequential component selected from said datapath library, said synthesizing step including generating IC expression trees from said hardware description language and synthesizing said netlist utilizing said IC expression trees and said datapath library; and

fabricating the integrated circuit as specified by said netlist.

A method executed by a computer under the control of a program, said computer including a memory for storing said program, said method comprising the steps of:

receiving in said computer a hardware description language description of an electronic circuit, said hardware description language description of said electronic circuit including circuit element descriptions and circuit connection descriptions, said circuit element descriptions defining datapath functions performed by said electronic circuit;

storing in said memory a library with a plurality of hardware description language datapath library elements that perform specified datapath operations;

parsing

said circuit element descriptions of said hardware description language description of said electronic circuit into circuit element intermediate register transfer level (RTL) descriptions, each of said circuit element intermediate RTL descriptions including a first list of pointers to data structures with multiple input and output ports and a second list of pointers to data structures representing logical or mathematical expressions, and

said hardware description language datapath library elements into datapath library element intermediate RTL descriptions, each of said datapath library element intermediate RTL descriptions including a first list of pointers to data structures representing input and output ports and a second list of pointers to data structures representing logical or mathematical expressions;

transforming

each of said circuit element intermediate RTL descriptions into a corresponding circuit element directed acyclic graph, thereby generating a set of circuit element directed acyclic graphs,

each of said datapath library element intermediate RTL descriptions into a corresponding datapath library element directed acyclic graph, thereby generating a set of datapath library element directed acyclic graphs;

synthesizing a datapath structure representing said electronic circuit, said synthesizing step including the step of matching said circuit element directed acyclic graphs with said datapath library element directed acyclic graphs to identify matched datapath library elements which perform said datapath functions defined by said circuit element descriptions; and

generating a netlist of circuit elements selected from said matched datapath library elements.

WO 9602038A

☐ 21. Document ID: US 20020049576 A1

L13: Entry 21 of 46 File: PGPB Apr 25, 2002

PGPUB-DOCUMENT-NUMBER: 20020049576
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020049576 A1

TITLE: Digital and analog mixed signal simulation using PLI API

PUBLICATION-DATE: April 25, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Meyer, Steven J.	Mill Valley	CA	US	

US-CL-CURRENT: 703/14; 703/4

ABSTRACT:

A system and method for analog and digital mixed mode simulation. The system and method simulates analog mixed signal (AMS) systems coded in one or a plurality of hardware description languages (HDLs) that describe digital subsystem, analog circuits, and mixed signal interface components. It implements and simulates AMS circuits using any standardized and specialized type of application programming interface (API) called a HDL programming language interface (PLIs). In it preferred embodiment, the system and method simulates systems coded in the popular Verilog-AMS HDL and legacy Spice HDLs. Utilization of the PLI allows for a much simplified and improved AMS simulation because the mixed mode engine implemented using the PLI invokes any commonly available digital simulator(s) for the digital engine(s) and any commonly available analog solver(s) for the analog engine(s). The system and method combines the accuracy of single kernel AMS simulation with the ease of construction and flexibility of data exchange AMS simulation.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 22. Document ID: US 6651225 B1

L13: Entry 22 of 46 File: USPT Nov 18, 2003

US-PAT-NO: 6651225
DOCUMENT-IDENTIFIER: US 6651225 B1

TITLE: Dynamic evaluation logic system and method

DATE-ISSUED: November 18, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lin; Sharon Sheau-Pyng	Cupertino	CA		
Tseng; Ping-Sheng	Sunnyvale	CA		
Chang; Chwen-Cher	Fremont	CA		
Hwang; Su-Jen	Los Altos	CA		

US-CL-CURRENT: 716/4; 716/1

ABSTRACT:

In a verification system, a dynamic logic evaluation system and method dynamically calculates the minimum evaluation time for each input. Thus, this system and method will remove the performance burden that a fixed and statically calculated evaluation time would introduce. By dynamically calculating different evaluation times based on the input, 99% of the inputs will not be delayed for the sake of 1% of the inputs that actually need the worst possible evaluation time. The dynamic logic evaluation system and method comprises a global control unit coupled to a propagation detector, where the propagation detector is placed in each FPGA chip. The propagation detector in the FPGA chip alerts the global control unit of any input data that is currently propagating within the FPGA chips. A master clock controls the operation of this dynamic evaluation system and method. As long as any input data is propagating, the global control unit will prevent the next input from being provided to the FPGA chips for evaluation. Once the output has stabilized, the global control unit will then instruct the system to accept and process the next set of input data. Thus, the global control unit in conjunction with the propagation detectors can dynamically provide varying evaluation time periods based on the needs of the input data. Whether the system needs longer or shorter evaluation times, the system will dynamically adjust the amount of time necessary to properly process that input and then move on to the next evaluation time for the next set of inputs.

20 Claims, 107 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 94

☐ 23. Document ID: US 6618839 B1

L13: Entry 23 of 46 File: USPT Sep 9, 2003

US-PAT-NO: 6618839
DOCUMENT-IDENTIFIER: US 6618839 B1

TITLE: Method and system for providing an electronic system design with enhanced debugging capabilities

DATE-ISSUED: September 9, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Beardslee; John Mark	Menlo Park	CA		
Schubert; Nils Endric	Sunnyvale	CA		
Perry; Douglas L.	San Ramon	CA		

US-CL-CURRENT: 716/4; 714/735, 716/5, 716/6

ABSTRACT:

Techniques and systems for analysis, diagnosis and debugging fabricated hardware designs at a Hardware Description Language (HDL) level are described. Although the hardware designs (which were designed in HDL) have been fabricated in integrated circuit products with limited input/output pins, the techniques and systems enable

the hardware designs within the integrated circuit products to be comprehensively analyzed, diagnosed, and debugged at the HDL level at speed. The ability to debug hardware designs at the HDL level facilitates correction or adjustment of the HDL description of the hardware designs.

35 Claims, 21 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 26

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments

KWIC | Draw Desc | Image

☐ 24. Document ID: US 6606588 B1

L13: Entry 24 of 46 File: USPT Aug 12, 2003

US-PAT-NO: 6606588
DOCUMENT-IDENTIFIER: US 6606588 B1

TITLE: Design apparatus and a method for generating an implementable description of a digital system

DATE-ISSUED: August 12, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schaumont; Patrick	Wijgmaal			BE
Vernalde; Serge	Heverlee			BE
Cockx; Johan	Pellenberg			BE

US-CL-CURRENT: 703/15; 716/18, 716/5, 716/7, 717/108

ABSTRACT:

The present invention is a design apparatus compiled on a computer environment for generating from a behavioral description of a system comprising at least one digital system part, an implementable description for said system, said behavioral description being represented on said computer environment as a first set of objects with a first set of relations therebetween, said implementable description being represented on said computer environment as a second set of objects with a second set of relations therebetween, said first and second set of objects being part of a design environment.

45 Claims, 29 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 25

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments

KWIC | Draw Desc | Image

☐ 25. Document ID: US 6581191 B1

L13: Entry 25 of 46 File: USPT Jun 17, 2003

US-PAT-NO: 6581191
DOCUMENT-IDENTIFIER: US 6581191 B1

TITLE: Hardware debugging in a hardware description language

DATE-ISSUED: June 17, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schubert; Nils Endric	Sunnyvale	CA		
Beardslee; John Mark	Menlo Park	CA		
Perry; Douglas L.	San Ramon	CA		

US-CL-CURRENT: 716/4; 703/14, 716/17

ABSTRACT:

Techniques and systems for analysis, diagnosis and debugging fabricated hardware designs at a Hardware Description Language (HDL) level are described. Although the hardware designs (which were designed in HDL) have been fabricated in integrated circuit products with limited input/output pins, the techniques and systems enable the hardware designs within the integrated circuit products to be comprehensively analyzed, diagnosed, and debugged at the HDL level at speed. The ability to debug hardware designs at the HDL level facilitates correction or adjustment of the HDL description of the hardware designs.

41 Claims, 28 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 26

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

---

☐ 26. Document ID: US 6421818 B1

L13: Entry 26 of 46 File: USPT Jul 16, 2002

US-PAT-NO: 6421818
DOCUMENT-IDENTIFIER: US 6421818 B1

TITLE: Efficient top-down characterization method

DATE-ISSUED: July 16, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dupenloup; Guy	Marly-le-Roi			FR
Cleereman; Kevin Christopher	Mounds View	MN		

US-CL-CURRENT: 716/18; 716/3

ABSTRACT:

A method of efficiently characterizing modules of an integrated circuit (IC) design using a logic synthesis tool comprising the steps of defining a list of instances of the modules to characterize, and characterizing entire modules of said list of instances of the modules using a single invocation of characterize command of the logic synthesis tool.

15 Claims, 57 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 38

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments KWIC | Draw Desc | Image

---

☐ 27. Document ID: US 6421808 B1

L13: Entry 27 of 46 File: USPT Jul 16, 2002

US-PAT-NO: 6421808
DOCUMENT-IDENTIFIER: US 6421808 B1
**** See image for Certificate of Correction ****

TITLE: Hardware design language for the design of integrated circuits

DATE-ISSUED: July 16, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
McGeer; Patrick C.	Orinda	CA		
Cheng; Szu-Tsung	El Cerrito	CA		
Meyer; Michael J.	Palo alto	CA		
Scaglia; Patrick	Saratoga	CA		

US-CL-CURRENT: 716/1; 703/14

ABSTRACT:

A hardware design language V++ is described. V++ provides an automatically designed and implemented communications protocol, embedded by a compiler in the design itself. This protocol permits transparent, automatic communication between modules in a hardware design. The protocol generalizes current design practice and impacts neither the cycle time, nor the area, of a typical system. Incorporating this protocol in the language itself frees the designer from the task of writing communications code, and ensures that two communicating modules follow the same low-level protocol. In V++ each program is directly interpreted as a network of communicating finite state machines. The composition of two V++ programs is a V++ program, with well-defined, deterministic semantics.

26 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 19

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 28. Document ID: US 6389379 B1

L13: Entry 28 of 46 File: USPT May 14, 2002

US-PAT-NO: 6389379
DOCUMENT-IDENTIFIER: US 6389379 B1

TITLE: Converification system and method

DATE-ISSUED: May 14, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lin; Sharon Sheau-Pyng	Cupertino	CA		
Tseng; Ping-Sheng	Sunnyvale	CA		

US-CL-CURRENT: 703/14; 703/23, 716/5

ABSTRACT:

The coverification system includes a reconfigurable computing system (hereinafter "RCC computing system") and a reconfigurable computing hardware array (hereinafter "RCC hardware array"). In some embodiments, the target system and the external I/O

devices are not necessary since they can be modeled in software. In other embodiments, the target system and the external I/O devices are actually coupled to the coverification system to obtain speed and use actual data, rather than simulated test bench data. The RCC computing system contains a CPU and memory for processing data for modeling the entire user design in software. The RCC computing system also contains clock logic (for clock edge detection and software clock generation), test bench processes for testing the user design, and device models for any I/O device that the user decides to model in software instead of using an actual physical I/O device. The user may decide to use actual I/O devices as well as modeled I/O devices in one debug session. The software clock is used as the external clock source for the target system and the external I/O devices to synchronize all data that is delivered between the coverification system and the external interface. The coverification system contains a control logic that provides traffic control between: (1) the RCC computing system and the RCC hardware array, and (2) the external interface (which are coupled to the target system and the external I/O devices) and the RCC hardware array. Because the RCC computing system has the model of the entire design in software, including that portion of the user design modeled in the RCC hardware array, the RCC computing system must also have access to all data that passes between the external interface and the RCC hardware array. The control logic ensures that the RCC computing system has access to these data. Pointers are used to latch data from the RCC computing system and the external interface to the internal nodes of the hardware model in the RCC hardware array. Pointers are also used to deliver data from the internal nodes of the hardware model to the RCC computing system and the external interface. Even if the data from the internal nodes of the hardware model is intended for the external interface, the RCC computing system must also be able to access this data as well.

17 Claims, 89 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 80

☐ 29. Document ID: US 6378123 B1

L13: Entry 29 of 46 File: USPT Apr 23, 2002

US-PAT-NO: 6378123
DOCUMENT-IDENTIFIER: US 6378123 B1

TITLE: Method of handling macro components in circuit design synthesis

DATE-ISSUED: April 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dupenloup; Guy	Marly-le-Roi			FR

US-CL-CURRENT: 716/18; 716/2

ABSTRACT:

A method of synthesizing integrated circuit (IC) design having DesignWare components comprising the steps of initially mapping DesignWare components, revising DesignWare component structure, ungrouping DesignWare components, and re-synthesizing DesignWare components. The step of initially mapping is performed using elaborate command and compile command of a logic synthesis tool. The step of ungrouping DesignWare components involves dissolving DesignWare modules to be merged with surrounding logic.

11 Claims, 57 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 38

☐ 30. Document ID: US 6324678 B1

L13: Entry 30 of 46 File: USPT Nov 27, 2001

US-PAT-NO: 6324678
DOCUMENT-IDENTIFIER: US 6324678 B1

TITLE: Method and system for creating and validating low level description of electronic design

DATE-ISSUED: November 27, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dangelo; Carlos	Los Gatos	CA		
Deeley; Richard	San Jose	CA		
Nagasamy; Vijay	Union City	CA		
Vafai; Manoucher	Los Gatos	CA		

US-CL-CURRENT: 716/18; 716/11, 716/5, 716/7, 716/8

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications is disclosed. The methodology uses a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially a series of transformations operating upon various levels of design representations. At each level, the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. A matrix of milestones (goals in the design activity) is defined by degree of complexity (level of abstraction) of a design and for progressive stages (levels) of design activity (from concept through implementation). The milestones are defined using continuous refinement, and the design activity proceeds towards subsequent milestones. As milestones are achieved, previous design activity becomes unalterable. A feasibility stage is key to convergence of the process. Single level or multi-level estimators determine the direction of the process.

23 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments
KWIC | Draw Desc | Image

☐ 31. Document ID: US 6295636 B1

L13: Entry 31 of 46 File: USPT Sep 25, 2001

US-PAT-NO: 6295636

DOCUMENT-IDENTIFIER: US [redacted]95636 B1

TITLE: RTL analysis for improved logic synthesis

DATE-ISSUED: September 25, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dupenloup; Guy	Marly-le-Roi			FR

US-CL-CURRENT: 716/18; 716/2

ABSTRACT:

A method of generating synthesis scripts to synthesize integrated circuit (IC) designs in RTL level description into gate-level description comprising the steps of identifying hardware elements in the RTL code, determining key pins for each of said identified hardware elements, extracting design structure and hierarchy from the RTL code, generating script to cause a logic synthesis tool to apply bottom-up synthesis to modules and sub-modules of the IC design, generating script to cause a logic synthesis tool to apply top-down characterization to modules and sub-modules of the IC design and generating script to cause a logic synthesis tool to repeat said bottom-up and said top-down applications until certain predetermined constraints are satisfied.

14 Claims, 57 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 38

☐ 32. Document ID: US 6292931 B1

L13: Entry 32 of 46 File: USPT Sep 18, 2001

US-PAT-NO: 6292931
DOCUMENT-IDENTIFIER: US 6292931 B1

TITLE: RTL analysis tool

DATE-ISSUED: September 18, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dupenloup; Guy	Marly-le-Roi			FR

US-CL-CURRENT: 716/18; 716/2, 716/3, 716/7

ABSTRACT:

A method of determining circuit characteristics of an integrated circuit design defined by RTL code, said method comprising the steps of identifying hardware elements in the RTL code, determining key pins for said identified hardware elements, and extracting critical design structure from the RTL code. The hardware elements identified include flipflops, latches, tristate buffers, bidirectional buffers and memories. The critical design structures include design hierarchy and nets, including clock nets, multiply-driven nets, reset nets, and RAM write enable nets.

35 Claims, 57 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 38

33. Document ID: US 6289491 B1

L13: Entry 33 of 46 File: USPT Sep 11, 2001

US-PAT-NO: 6289491
DOCUMENT-IDENTIFIER: US 6289491 B1

TITLE: Netlist analysis tool by degree of conformity

DATE-ISSUED: September 11, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dupenloup; Guy	Marly-le-Roi			FR

US-CL-CURRENT: 716/5

ABSTRACT:

A method of determining circuit characteristics of an integrated circuit design as defined by a generic netlist comprising the steps of identifying hardware elements in the generic netlist, determining key characteristics for each of said identified hardware elements, determining interconnections of said identified hardware elements, and detecting the degree of conformity of said identified hardware elements, said key characteristics, and said interconnections to predetermined configurations. The systems further identifies all cells in the generic netlist, determines for each cell the type of cell, accumulates cell types and cell type counts, and notifies an operator of said accumulated values.

17 Claims, 57 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 38

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

34. Document ID: US 6205572 B1

L13: Entry 34 of 46 File: USPT Mar 20, 2001

US-PAT-NO: 6205572
DOCUMENT-IDENTIFIER: US 6205572 B1

TITLE: Buffering tree analysis in mapped design

DATE-ISSUED: March 20, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dupenloup; Guy	Marly le-Roi			FR

US-CL-CURRENT: 716/5; 716/2

ABSTRACT:

A method of determining circuit characteristics of buffering tree nets of an integrated circuit (IC) design comprising the steps of determining source pins of

the nets of the buffering tree, determining fanout of each of said source pins, determining active edges and active levels of each of said source pins, and presenting said source pins, said fanout, and said active edge on a report.

19 Claims, 57 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 37

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 35. Document ID: US 6134516 A

L13: Entry 35 of 46 File: USPT Oct 17, 2000

US-PAT-NO: 6134516
DOCUMENT-IDENTIFIER: US 6134516 A

TITLE: Simulation server system and method

DATE-ISSUED: October 17, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wang; Steven	Cupertino	CA		
Tseng; Ping-Sheng	Sunnyvale	CA		
Lin; Sharon Sheau-Pyng	Cupertino	CA		
Tsay; Ren-Song	Palo Alto	CA		
Sun; Richard Yachyang	San Jose	CA		
Shen; Quincy Kun-Hsu	Union City	CA		
Tsai; Mike Mon Yen	Los Altos Hills	CA		

US-CL-CURRENT: 703/27; 703/20, 712/20

ABSTRACT:

The SEmulation system provides four modes of operation: (1) Software Simulation, (2) Simulation via Hardware Acceleration, (3) In-Circuit Emulation (ICE), and (4) Post-Simulation Analysis. At a high level, the present invention may be embodied in each of the above four modes or various combinations of these modes. At the core of these modes is a software kernel which controls the overall operation of this system. The main control loop of the kernel executes the following steps: initialize system, evaluate active test-bench processes/components, evaluate clock components, detect clock edge, update registers and memories, propagate combinational components, advance simulation time, and continue the loop as long as active test-bench processes are present. A Simulation server in accordance with an embodiment of the present invention allows multiple users to access the same reconfigurable hardware unit to effectively simulate and accelerate the same or different user designs in a time-shared manner in both a network and a non-network environment. The server provides the multiple users or processes to access the reconfigurable hardware unit for acceleration and hardware state swapping purposes. The Simulation server includes the scheduler, one or more device drivers, and the reconfigurable hardware unit. The scheduler in the Simulation server is based on a preemptive round robin algorithm. The server scheduler includes a simulation job queue table, a priority sorter, and a job swapper.

27 Claims, 78 Drawing figures
Exemplary Claim Number: 16
Number of Drawing Sheets: 68

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 36. Document ID: US 6051031 A

L13: Entry 36 of 46 File: USPT Apr 18, 2000

US-PAT-NO: 6051031
DOCUMENT-IDENTIFIER: US 6051031 A

TITLE: Module-based logic architecture and design flow for VLSI implementation

DATE-ISSUED: April 18, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Shubat; Alexander	Fremont	CA		
Kablanian; Adam	San Jose	CA		
Duvalyan; Vardan	Sunnyvale	CA		

US-CL-CURRENT: 716/3; 703/15

ABSTRACT:

A new design methodology which utilizes a module-based architecture is used to implement customized VLSI designs. In accordance with this invention, the module-based architecture comprises a number of Matrix Transistor Logic (MTL) modules. Each MTL module has a control input buffer section, an output stage section, and a matrix array section. The matrix array section implements logic functions using Pass Transistor Logic technology. Three variables, each of which place a different constraint on the MTL modules, are used in an automated design procedure to implement the MTL modules.

74 Claims, 30 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 30

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments

KWIC | Draw Desc | Image

☐ 37. Document ID: US 6026230 A

L13: Entry 37 of 46 File: USPT Feb 15, 2000

US-PAT-NO: 6026230
DOCUMENT-IDENTIFIER: US 6026230 A

TITLE: Memory simulation system and method

DATE-ISSUED: February 15, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lin; Sharon Sheau-Pyng	Cupertino	CA		
Tseng; Ping-Sheng	Sunnyvale	CA		

US-CL-CURRENT: 703/13

ABSTRACT:

The SEmulation system provides four modes of operation: (1) Software Simulation, (2) Simulation via Hardware Acceleration, (3) In-Circuit Emulation (ICE), and (4) Post-Simulation Analysis. At a high level, the present invention may be embodied in

each of the above four modes or various combinations of these modes. At the core of these modes is a software kernel which controls the overall operation of this system. The main control loop of the kernel executes the following steps: initialize system, evaluate active test-bench processes/components, evaluate clock components, detect clock edge, update registers and memories, propagate combinational components, advance simulation time, and continue the loop as long as active test-bench processes are present. The Memory Mapping aspect of the invention provides a structure and scheme where the numerous memory blocks associated with the user's design is mapped into the SRAM memory devices in the Simulation system instead of inside the logic devices, which are used to configure and model the user's design. The Memory Mapping or Memory Simulation system includes a memory state machine, an evaluation state machine, and their associated logic to control and interface with: (1) the main computing system and its associated memory system, (2) the SRAM memory devices coupled to the FPGA buses in the Simulation system, and (3) the FPGA logic devices which contain the configured and programmed user design that is being debugged.

20 Claims, 76 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 68

☐ 38. Document ID: US 6009256 A

L13: Entry 38 of 46 File: USPT Dec 28, 1999

US-PAT-NO: 6009256
DOCUMENT-IDENTIFIER: US 6009256 A

TITLE: Simulation/emulation system and method

DATE-ISSUED: December 28, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tseng; Ping-Sheng	Sunnyvale	CA		
Lin; Sharon Sheau-Pyng	Cupertino	CA		
Shen; Quincy Kun-Hsu	Union City	CA		
Sun; Richard Yachyang	San Jose	CA		
Tsai; Mike Mon Yen	Los Altos Hills	CA		
Tsay; Ren-Song	Palo Alto	CA		
Wang; Steven	Cupertino	CA		

US-CL-CURRENT: 703/13; 703/23

ABSTRACT:

The SEmulation system provides four modes of operation: (1) Software Simulation, (2) Simulation via Hardware Acceleration, (3) In-Circuit Emulation (ICE), and (4) Post-Simulation Analysis. At a high level, the present invention may be embodied in each of the above four modes or various combinations of these modes. At the core of these modes is a software kernel which controls the overall operation of this system. The main control loop of the kernel executes the following steps: initialize system, evaluate active test-bench processes/components, evaluate clock components, detect clock edge, update registers and memories, propagate combinational components, advance simulation time, and continue the loop as long as active test-bench processes are present. Each mode or combination of modes provides the following main features or combinations of main features: (1) switching among modes, manually or automatically; (2) compilation process to generate software models and hardware models; (3) component type analysis for generating hardware models; (4) software clock set-up to avoid race conditions through, in one embodiment, gated

clock logic analysis and gated data logic analysis; (5) software clock implementation through, in one embodiment, clock edge detection in the software model to trigger an enable signal in the hardware model, send signal from the primary clock to the clock input of the clock edge register in the hardware model via the gated clock logic, send a clock enable signal to the enable input of the hardware model's register, send data from the primary clock register to the hardware model's register via the gated data logic, and reset the clock edge register disabling the clock enable signal to the enable input of the hardware model's registers; (6) log selective data for debug sessions and post-simulation analysis; and (7) combinational logic regeneration.

45 Claims, 41 Drawing figures
Exemplary Claim Number: 24
Number of Drawing Sheets: 39

☐ 39. Document ID: US 5870588 A

L13: Entry 39 of 46 File: USPT Feb 9, 1999

US-PAT-NO: 5870588
DOCUMENT-IDENTIFIER: US 5870588 A

TITLE: Design environment and a design method for hardware/software co-design

DATE-ISSUED: February 9, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rompaey; Karl Van	Heist-ap-den-Berg			BE
Verkest; Diederik	Kapellen			BE
Vanhoof; Jan	Wijmaal			BE
Lin; Bill	Leuven			BE
Bolsens; Ivo	Schoten			BE
De Man; Hugo	Leuven			BE

US-CL-CURRENT: 703/13; 709/316

ABSTRACT:

A hardware and software co-design environment and design methodology based on a data-model that allows one to specify, simulate, and synthesize heterogeneous hardware and software architectures from a heterogeneous specification. The environment and methodology of the invention allow for the interactive synthesis of hardware and software interfaces. The environment defines primitive objects to represent a specification of an essentially digital system. The primitive objects are defined by describing the specification of the system in one or more processes, each process representing a functional aspect of the system. Further, each of the processes have ports which are connected to ports of other processes with a channel. The ports structure communication between the processes.

46 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

☐ 40. Document ID: US 5870308 A

L13: Entry 40 of 46 File: USPT Feb 9, 1999

US-PAT-NO: 5870308
DOCUMENT-IDENTIFIER: US 5870308 A

TITLE: Method and system for creating and validating low-level description of electronic design

DATE-ISSUED: February 9, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dangelo; Carlos	Los Gatos	CA		
Nagasamy; Vijay	Union City	CA		
Ponukumati; Vijayanand	Sunnyvale	CA		

US-CL-CURRENT: 716/18; 716/2

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications is disclosed. The methodology uses a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially a series of transformations operating upon various levels of design representations. At each level, the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. Techniques are described for estimating ancillary parameters of the device (such as device cost, production speed, production lead time, etc.), at early, high level stages of the design process (e.g., at the system, behavioral, and register transfer level stages). The techniques can be applied to optimize the design characteristics other than measurable physical characteristics, such as those deriving from project time and cost constraints.

37 Claims, 21 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 19

Search Results - Record(s) 41 through 46 of 46 returned.

☐ 41. Document ID: US 5703789 A

L13: Entry 41 of 46 File: USPT Dec 30, 1997

US-PAT-NO: 5703789
DOCUMENT-IDENTIFIER: US 5703789 A

TITLE: Test ready compiler for design for test synthesis

DATE-ISSUED: December 30, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Beausang; James	Mountain View	CA		
Walker; Robert	Boulder	CO		

US-CL-CURRENT: 716/4; 716/5

ABSTRACT:

A computer implemented process and system for providing a test ready (TR) compiler with specific information regarding the impact of added scannable cells and resources on its mission mode design. In so doing, the TR compiler optimizes more effectively for added test resources (e.g., scannable cells and other scan routing resources) so that predetermined performance and design related constraints of the mission mode design are maintained. The TR compiler translates generic sequential cells into technology dependent non-scan cells. In the TR compiler, during replacement, scannable memory cells are used in place of these non-scan memory cells specified within the mission mode circuitry. In this way, the TR compiler is informed of the characteristics of the scannable memory cells during optimization. For test, the scannable memory cells are chained to each other to form chain chains of sequential cells. To account for chaining during compile, the TR compiler provides output driven loopback connections to simulate electrical characteristics of the chain during compile. In the above implementation, the TR compiler can efficiently provide translation of an HDL description with test implementations into a gate level netlist. With the addition of certain information regarding the test implementation (e.g., scan replacement is done and loopback connections are added), the TR compiler of the present invention can better optimize the overall layout for the addition of the test resources.

36 Claims, 33 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 19

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 42. Document ID: US 5598344 A

L13: Entry 42 of 46 File: USPT Jan 28, 1997

US-PAT-NO: 5598344
DOCUMENT-IDENTIFIER: US 5598344 A

TITLE: Method and system for creating, validating, and s[illegible]ling structural description of electronic device

DATE-ISSUED: January 28, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dangelo; Carlos	Los Gatos	CA		
Mintz; Doron	Sunnyvale	CA		
Vafai; Manouchehr	Los Gatos	CA		

US-CL-CURRENT: 716/18; 716/4

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications. The methodology uses a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially a series of transformations operating upon various levels of design representations. At each level, the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. Techniques for scaling of a model design to provide a scaled design are provided whereby parameters of a model design such as size, circuit complexity, interconnection density, number of I/O connections, etc., can be scaled to produce a scaled version of the design. The scaling techniques employ multi-level hierarchical module replication to produce fully-functional scaled designs which closely match the function of the model design. Test vectors for the scaled designs can be readily obtained by altering test vectors for the model design to account for the replicated modules.

19 Claims, 23 Drawing figures
Exemplary Claim Number: 14
Number of Drawing Sheets: 20

☐ 43. Document ID: US 5572436 A

L13: Entry 43 of 46 File: USPT Nov 5, 1996

US-PAT-NO: 5572436
DOCUMENT-IDENTIFIER: US 5572436 A

TITLE: Method and system for creating and validating low level description of electronic design

DATE-ISSUED: November 5, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dangelo; Carlos	Los Gatos	CA		
Nagasamy; Vijay	Union City	CA		
Ponukumati; Vijayanand	Sunnyvale	CA		

US-CL-CURRENT: 716/18; 703/13, 716/11

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications using a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially a series of transformations operating upon various levels of design representations. At each level, the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. Techniques are provided for estimating ancillary parameters of the device (such as device cost, production speed, production lead time, etc.), at early, high level stages of the design process (e.g., at the system, behavioral, and register transfer level stages). The techniques can be applied to optimize the design characteristics other than measurable physical characteristics, such as those deriving from project time and cost constraints.

19 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

☐ 44. Document ID: US 5555201 A

L13: Entry 44 of 46 File: USPT Sep 10, 1996

US-PAT-NO: 5555201
DOCUMENT-IDENTIFIER: US 5555201 A

TITLE: Method and system for creating and validating low level description of electronic design from higher level, behavior-oriented description, including interactive system for hierarchical display of control and dataflow information

DATE-ISSUED: September 10, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dangelo; Carlos	Los Gatos	CA		
Watkins; Daniel	Los Altos	CA		
Mintz; Doron	Sunnyvale	CA		

US-CL-CURRENT: 716/1; 703/13, 716/18

ABSTRACT:

A technique for hierarchical display of control and dataflow graphs allowing a user to view hierarchically filtered control and dataflow information related to a design. The technique employs information inherent in the design description and information derived from design synthesis to identify "modules" of the design and design hierarchy. The user can specify a level of detail to be displayed for any design element or group of design elements. Any CDFG (control and dataflow graph) object can be "annotated" with a visual attribute or with text to indicate information about the design elements represented by the object. For example, block

size, interior color, border color, line thickness, line style, etc., can be used to convey quantitative or qualitative information about a CDFG object. Examples of information which can be used to "annotate" objects include power dissipation, propagation delay, the number of HDL statement represented, circuit area, number of logic gates, etc. The user is able to expand and/or compress CDFG blocks either "in-place" on a higher level CDFG display or to be displayed in isolation. Simulation-related data can also be used to annotate the CDFG. By viewing CDFG's (particularly annotated CDFG's) for a variety of trial designs, a problem-solving user can gain quick insight into the effects and effectiveness of various design choices.

24 Claims, 65 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 49

☐ 45. Document ID: US 5541850 A

L13: Entry 45 of 46 File: USPT Jul 30, 1996

US-PAT-NO: 5541850
DOCUMENT-IDENTIFIER: US 5541850 A

TITLE: Method and apparatus for forming an integrated circuit including a memory structure

DATE-ISSUED: July 30, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Vander Zanden; Nels B.	Mountain View	CA		
Mahmood; Mossaddeq	San Jose	CA		

US-CL-CURRENT: 716/18; 716/19, 716/5, 716/8

ABSTRACT:

A set of circuit specifications including an internal memory structure is developed and then described in a hardware description language that is entered into a computer system. The circuit description is then synthesized on the computer to form a netlist to specify the circuit. From this netlist, an integrated circuit is produced on a semiconductor die, which is then packaged for use. A method for synthesizing a netlist from a hardware description including an internal memory structure includes converting the hardware description into an internal signal list, which contains an indication of the presence of an internal memory structure in the described circuit. For each memory structure indicated, synthesis is performed using a memory cell library, and compatibility between the hardware description for the circuit and the internal memory structure specified is determined. When compatibility is found, the internal memory structure is instantiated into the netlist for the circuit. A central processing unit (CPU) is connected to a keyboard used to input a hardware description of a circuit. Further included is a hardware description processor implemented on the CPU that creates mask generation data for use with a mask generator to form an integrated circuit. An internal memory structure as described in the hardware description of the circuit is thereby included in the circuit.

43 Claims, 20 Drawing figures
Exemplary Claim Number: 37
Number of Drawing Sheets: 15

☐ 46. Document ID: US 5541849 A

L13: Entry 46 of 46 File: USPT Jul 30, 1996

US-PAT-NO: 5541849
DOCUMENT-IDENTIFIER: US 5541849 A

TITLE: Method and system for creating and validating low level description of electronic design from higher level, behavior-oriented description, including estimation and comparison of timing parameters

DATE-ISSUED: July 30, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rostoker; Michael D.	Boulder Creek	CA		
Dangelo; Carlos	Los Gatos	CA		
Mintz; Doron	Sunnyvale	CA		

US-CL-CURRENT: 716/18; 716/19, 716/5

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications. The methodology uses a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially a series of transformations operating upon various levels of design representations. At each level, the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. Techniques are provided for estimating design performance, from behavioral/functional descriptions. Given a behavioral or a block diagram description of data flow in a design, pin-to-pin timing and minimum clock cycle for the design can be estimated accurately. An RTL description may thus be synthesized from a behavioral description such that timing constraints imposed at the behavioral level are achieved. The timing of a synthesized design is estimated, and the design is re-synthesized until a design is arrived at that meets timing constraints imposed at a higher level.

8 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Display Format: [-] [Change Format]

Previous Page Next Page

Search Results - Record(s) 1 through 4 of 4 returned.

☐ 1. Document ID: US 6625797 B1

L19: Entry 1 of 4 File: USPT Sep 23, 2003

US-PAT-NO: 6625797
DOCUMENT-IDENTIFIER: US 6625797 B1

TITLE: Means and method for compiling high level software languages into algorithmically equivalent hardware representations

DATE-ISSUED: September 23, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Edwards; Stephen G.	Woodbine	MD		
Harris; Jonathan Craig	Crofton	MD		
Jensen; James E.	Ellicott City	MD		
Kollegger; Andreas Benno	Baltimore	MD		
Miller; Ian David	Charlotte	NC		
Sunderland Schanck; Christopher Robert	Ellicott City	MD		
Davis; Donald J.	Ellicott City	MD		

US-CL-CURRENT: 716/18; 702/108, 702/125, 702/85, 716/1, 716/16, 716/4, 716/6

ABSTRACT:

The compilation of a high-level software-based description of an algorithm into efficient digital hardware implementation(s) is addressed. This is done through the definition of new semantics for software constructs with respect to hardware implementations. This approach allows a designer to work at a high level of abstraction, while the semantic model can be used to infer the resulting hardware implementation. These semantics are interpreted through the use of a compilation tool that analyzes the software description to generate a control and data flow graph. This graph is then the intermediate format used for optimizations, transformations and annotations. The resulting graph is then translated to either a register transfer level or a netlist-level description of the hardware implementation.

45 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

☐ 2. Document ID: US 5910897 A

L19: Entry 2 of 4 File: USPT Jun 8, 1999

US-PAT-NO: 5910897
DOCUMENT-IDENTIFIER: US 5910897 A

TITLE: Specification [redacted] design of complex digital sys[redacted]s

DATE-ISSUED: June 8, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dangelo; Carlos	Los Gatos	CA		
Nagasamy; Vijay	Union City	CA		

US-CL-CURRENT: 716/19; 716/18, 716/2, 716/4

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications is disclosed. The methodology uses a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially a series of transformations operating upon various levels of design representations. At each level the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a-high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. The methodology further includes an automated interactive, iterative technique for creating a system-level specification in a directly-executable formal specification language. This technique makes use of formal verification and feasibility analysis techniques to iteratively refine the specification prior to implementation. This iterative refinement eliminates many ambiguities and inconsistencies from the specification, and ensures that there is at least one realizable implementation of the specification. The formal verification techniques are further employed to ensure that as the design progresses, compliance with the specification is maintained, and that any specification change is reflected and accounted for, both system-wide and implementation-wide.

20 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15

☐ 3. Document ID: US 5541849 A

L19: Entry 3 of 4 File: USPT Jul 30, 1996

US-PAT-NO: 5541849
DOCUMENT-IDENTIFIER: US 5541849 A

TITLE: Method and system for creating and validating low level description of electronic design from higher level, behavior-oriented description, including estimation and comparison of timing parameters

DATE-ISSUED: July 30, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rostoker; Michael D.	Boulder Creek	CA		
Dangelo; Carlos	Los Gatos	CA		
Mintz; Doron	Sunnyvale	CA		

US-CL-CURRENT: 716/18; 716/19, 716/5

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications. The methodology uses a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially a series of transformations operating upon various levels of design representations. At each level, the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. Techniques are provided for estimating design performance, from behavioral/functional descriptions. Given a behavioral or a block diagram description of data flow in a design, pin-to-pin timing and minimum clock cycle for the design can be estimated accurately. An RTL description may thus be synthesized from a behavioral description such that timing constraints imposed at the behavioral level are achieved. The timing of a synthesized design is estimated, and the design is re-synthesized until a design is arrived at that meets timing constraints imposed at a higher level.

8 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

☐ 4. Document ID: US 5493508 A

L19: Entry 4 of 4 File: USPT Feb 20, 1996

US-PAT-NO: 5493508
DOCUMENT-IDENTIFIER: US 5493508 A

TITLE: Specification and design of complex digital systems

DATE-ISSUED: February 20, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dangelo; Carlos	Los Gatos	CA		
Nagasamy; Vijay	Union City	CA		

US-CL-CURRENT: 716/5

ABSTRACT:

A methodology for generating structural descriptions of complex digital devices from high-level descriptions and specifications is disclosed. The methodology uses a systematic technique to map and enforce consistency of the semantics imbedded in the intent of the original, high-level descriptions. The design activity is essentially

a series of transformations operating upon various levels of design representations. At each level, the intended meaning (semantics) and formal software manipulations are captured to derive a more detailed level describing hardware meeting the design goals. Important features of the methodology are: capturing the users concepts, intent, specification, descriptions, constraints and trade-offs; architectural partitioning; what-if analysis at a high level; sizing estimation; timing estimation; architectural trade-off; conceptual design with implementation estimation; and timing closure. The methodology includes using estimators, based on data gathered over a number of realized designs, for partitioning and evaluating a design prior to logic synthesis. From the structural description, a physical implementation of the device is readily realized. The methodology further includes an automated interactive, iterative technique for creating a system-level specification in a directly-executable formal specification language. This technique makes use of formal verification and feasibility analysis techniques to iteratively refine the specification prior to implementation. This iterative refinement eliminates many ambiguities and inconsistencies from the specification, and ensures that there is at least one realizable implementation of the specification. The formal verification techniques are further employed to ensure that as the design progresses, compliance with the specification is maintained, and that any specification change is reflected and accounted for, both system-wide and implementation-wide.

4 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15

Terms	Documents
L16 and temporal	4

CiteSeer Find: HDL and mapping and node Documents Citations

Searching for **hdl and mapping and node**.
Restrict to: Header Title Order by: Citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
10 documents found. **Order: citations weighted by year.**

Virtual Wires: Overcoming Pin Limitations in FPGA-based Logic.. - Babb, Tessier (1993) (Correct) (45 citations)
including an integrated interface to compiled-code **HDL** simulation and implementation of virtual logic
APR stage and assumes a 50 percent APR **mapping** efficiency. Before compiling the two test
and downloads an emulation image onto a 9 FPGA **node** prototype board. Work on a scalable 16 **node** board
ftp.cag.lcs.mit.edu/virtual_wires/mit_student_1993.ps.Z

One or more of the query terms is very common - only partial results have been returned. Try Google (RI).

Time-Constrained Code Compaction for DSPs - Leupers, Marwedel (1997) (Correct) (8 citations)
extraction: A hardware description language (**HDL**) model of the target processor is analyzed and is
should be assigned to programmable processors, and **mapping** these pieces into executable, processorspecific
subexpressions in ETs are resolved heuristically by **node** duplication [7]Instruction-set extraction: A
ls12-www.cs.uni-dortmund.de/publications/papers/1997-ieee_tvlsi.ps.gz

Performance Driven Synthesis for Pass-Transistor Logic - Liu (1998) (Correct) (2 citations)
A logic design in a hardware description language (**HDL**) was synthesized by a logic/circuit synthesis tool
(logic optimization) stage and a technology-**mapping** stage. In the former stage, This research was
synthesis has targeted minimizing the number of BDD **nodes**. This strategy leads to smaller PTL
www.ece.utexas.edu/~tlliu/publications/vlsi99.ps.gz

The Design of a Time Warp Synchronized VHDL Simulation Kernel - Subramani (1998) (Correct) (1 citation)
is to make use of a Hardware Description Language (**HDL**) to model the system in software and then simulate
:97 Bibliography 99 iii List of Tables 6.1 **Mapping** between types and universal types :
:55 7.6 Internal **node** in source tree :
www.ececs.uc.edu/~paw/lab/theses/skrish.ps.gz

Verification of VLSI Circuits: Signal Value Modeling and HDL .. - Gambles, Windley (Correct)
of VLSI Circuits: Signal Value Modeling and **HDL** Translation Jody W. Gambles Microelectronics
of the **HDL** to an internal abstract syntax. ffi **Mapping** the abstract syntax to the **HDL**'s semantic
and pre-charge logic, it is common for circuit **nodes** to be driven from multiple circuit devices. These
lal.cs.byu.edu/pub/hol/lal-papers/sig.val.modeling.ps

An Optimized Design Flow for Fast FPGA-Based Rapid.. - Stohmann, Harbich.. (1998) (Correct)
Finally, Section 5 provides concluding remarks. **HDL** (RTL) Netlist (Gate-Level) Synthesis Placement /
by letting the basic design steps technology **mapping**, hierarchical partitioning, floorplanning and
even adders and multipliers -is represented by one **node** or in case of decomposed components by several
www.ims.uni-hannover.de/~harbich/public/paper/fpl98.ps.gz

A tool-set for simulating Altera-PLDs using VHDL - Andr'e Klindworth (1994) (Correct)
comes with an own hardware description language (**HDL**) named AHDL [3]Compared to standard **HDLs** like
gives the user a direct control over technology **mapping** and design partitioning. Both aspects let us
during simulation using the original (hierarchical) **node** identifiers. 4 Conclusions The tool-set as
tech-www.informatik.uni-hamburg.de/Paper/1994/FPL94/FPL94.ps.gz

High-level Architectural Simulation of the Torus Routing Chip - Lasse Natvig (1997) (Correct)
the hardware side, Hardware Description Languages (**HDLs**) have become important tools to cope with this
crucial for correct behaviour. The possibility of **mapping** the model into hardware at a later stage should
and reduces the load on the main processor of each **node**. The TRC is such a chip. The TRC provides routing
www.idt.ntnu.no/~lasse/publics/ivc97ln.ps

The Multigraph Modeling Tool - Childers, Apon, Hooper, Gordon, Dowdy (Correct)
via the Multigraph High-level Description Language (**HDL**) 1]From this description MMT generates an

on which the application executes, and 3) the **mapping** between the two, and produce as outputs flow graph shown in Figure 1(b) consists of actor **nodes** A i that are equivalent to the application's
www.vuse.vanderbilt.edu/~cac/mmt.ps

CiteSeer Find: HDL and mapping and control structu Documents Citations

Searching for **hdl and mapping and control structure**.
Restrict to: Header Title Order by: Citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
2 documents found. **Order: citations weighted by year.**

An Integrated Framework For High-Level Synthesis Of Self-Timed.. - Akella (1992) (Correct) (2 citations)
circuits. hopCP is a process-oriented concurrent **HDL** for the specification of asynchronous systems. It
:54 4.2.1 **Mapping** From Logical to Physical Channels :
: 44 3.3.1.6 Translating **Control Structure** :46 3.3.1.7
ftp.ece.ucdavis.edu/pub/akella/VAThesis.ps.Z

One or more of the query terms is very common - only partial results have been returned. Try Google (RI).

Using a Soft Core in a SoC Design: Experiences with.. - Dey, Panigrahi, Chen.. (2000) (Correct)
and some tooldependenthpenden description language (**HDL**) constructs tosynthOGRF th picoJava. 2. Handling
core provider delivers a gate-level implementation **mapped** to a particular techcular library, after
low-power management and design-for-test **structure control**.Th processor core also supports various
esdat.ucsd.edu/~lichen/esdat/paper/DnT_final.pdf

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP

TOP 20 WEB RESULTS out of about 429 (What's this?)

1. Books by Ben Cohen
... LANGUAGE ELEMENTS: Why **VHDL** for digital designs ... supported/unsupported constructs,
synthesis sensitivity rules ... wait" statement, **control structure**, signals versus ...
klabs.org/richcontent/software_content/vhdl/vhdl_cohen.html - 12k - Cached

2. Very High-Level **Synthesis** of Datapath and Control Structures for ... (PDF)
... Only loop bodies are considered for **synthesis**, there can ... The **control structure** must be simple, that is, we only ... implemented, as well as the **VHDL** code generation ...
www.isi.edu/~joonseok/PUBLICATIONS/cases99.pdf - View as html - More pages from this site

3. Citations: A Formal Methodology for Automated **Synthesis** of VLSI ...
... net is used to represent the **control structure** of a ... A Formal Methodology for Automated
Synthesis of VLSI ... Two Methods for Synthesizing **VHDL** Concurrent Processes ...
citeseer.nj.nec.com/context/155519/0 - 22k - Cached - More pages from this site

4. Target Compiler Technologies : Recent News
... but so far the route towards **VHDL synthesis** was missing ... for generating Verilog, in addition to **VHDL**. ... in efficient processors with a simple **control structure**. ...
www.retarget.com/press-000518.html - 14k - Cached

5. CS 397 Chien September 7, 1994 (POSTSCRIPT)
... **Synthesis** &Optimization HDL. ... structure, INPUT/OUTPUT decls-**control structure** "names"
uDifferences ... and Event Cancellation, resolution in **VHDL**-Static Allocation ...
www-csag.ucsd.edu/individual/achien/cs397/Lecture3.ps - View as html

6. Contents
... 3 Structural Model: 3.1 Data Path Model; 3.2 Controller Model: 3.2.1 Properties Preserved by the **Control Structure**. ... D.4 Modules. E BNF of **VHDL Synthesis** Subset. ...
www.ececs.uc.edu/~ddel/theses/jay_new/node1.html - 19k - Cached - More pages from this site

7. **vhdl** asic code design links models **vhdl**
... A multi-platform, multi-language IDE that adds **Control Structure** Diagrams to Java ... VHDLSynth
(1076.3) Home page The **VHDL Synthesis** working group (IEEE 1076.3) is ...
www.programming-x.com/programming-x/vhdl.html - 12k - Cached

8. CEG5010 Tutorial Week 1
... the functional correctness) step 6: **synthesis** it (get ... **Control Structure** Common control
structures include: if...then ... Synthesizable **VHDL** subset Only a very limit ...
appsrv.cse.cuhk.edu.hk/~ceg5010/tut/week1.htm - 7k - Cached

9. **Synthesis** of complex control structures from behavioral SystemC ... (PDF)
... hardware design lan- guages such as Verilog and **VHDL**. ... 4 **Synthesis** issues One of the
major limits encountered ... is, an ¦@9 - 4&# ¡¤4 **control structure** with 7 58 ...
sigda.org/Archives/ProceedingArchives/Date/papers/2003/date03/df_files/03d_6.pdf - View as html - More pages from this site

10. An algorithm for microcode compaction of **VHDL** behavioral ...
... microcode compaction of **VHDL** behavioral descriptions. ... MICROPROGRAMMING B.1.2 **Control**
Structure Performance Analysis ... Design Aids Subjects: Automatic **synthesis****. ...
portal.acm.org/citation.cfm?
id=62204&jmp=cit&coll=GUIDE&dl=ACM&CFID=11111111&CFTOKEN=2222222 - More pages from this site

11. Efficient Hardware Controller **Synthesis** for Synchronous Dataflow ...
... This paper concerns automatic hardware **synthesis** from data flow ... that contains a synthesizable
VHDL code ... synthesizes a clever **control structure**, cascaded counter ...
csdl.computer.org/comp/proceedings/isss/2000/0765/00/07650079abs.htm - 10k - Cached

12. Automatic production of controller specifications from control ...
... B. Hardware B.1 CONTROL STRUCTURES AND MICROPROGRAMMING B.1.2 **Control Structure**
Performance Analysis and Design Aids Subjects: Automatic **synthesis****. ...
dx.doi.org/10.1145/74382.74396 - 38k - Cached

13. HDL, **VHDL**, Verilog and FPGA Training from Esperan
... EDA EDA focussed tools, applications and Synopsys **synthesis** scripts, including a ... and
folding text editor with **VHDL** mode and **Control Structure** Diagram (CSD ...
www.esperan.com/resources.html - 36k - Cached

14. Electronics Weekly HyperActive - **VHDL** Tool Kit
... Synthesizing combinational logic from a **VHDL** process is ... In **synthesis** jargon, this scenario is referred to as ... depth of your nested **control structure** jungle, you ...
www.electronicsweekly.com/toolkits/vhdl/vhdl6_6.asp - 7k - Cached

15. The BPL Group Organization The BPL Group Organization (PDF)
... Bedside Monitors and Defibrillation System and **Control Structure** Various device ... USA
: PVR Reference Design Soft Networks – USA : **VHDL Synthesis** and Protocol ...
www.bpltelecom.com/bpltelecom/images/downloads/General-SESG_Presentation (Comprehensive).pdf - View as html

16. Synchronization Control-dependent synchronization Data- ... (PDF)
... of specification languages 3.4.1 **VHDL** 3.4.2 ... synchronization techniques, it is the **control structure** of the ... are intended for use by **synthesis** and verification ...
www.physics.utu.fi/ett/kurssi/xG612/2002/04_jansson.pdf - View as html

17. Efficient Integration of Behavioral **Synthesis** within Existing ... (PDF)
... is easier to understand because **synthesis** tries to keep the initial **control structure**. ... DLS:
A scheduling algorithm for high- level **synthesis** in **VHDL**", Proc ...
tima.imag.fr/sls/documents/Isss_woc.pdf - View as html - More pages from this site

18. EE 5325 – Hardware Modeling with **VHDL** (MICROSOFT POWERPOINT)
... To ensure control as **synthesis** proceeds, it is best to ... Parentheses are sometimes used to **control structure**. The behavioral components of the **VHDL** source, ie ...
www.utdallas.edu/~svd031000/EE5325/vhdl_lecture16.ppt - View as html

19. Efficient hardware controller **synthesis** for synchronous dataflow ... (PDF)
... This paper concerns automatic hardware **synthesis** from data ... that contains a synthesizable
VHDL code. ... synthesizes a clever **control structure**, cascaded counter ...
peace.snu.ac.kr/publications/data/14/tvlsi2002_hyunuk.pdf - View as html - More pages from this site

20. Microsoft PowerPoint - CSCE611(Fall02)-Lecture11.ppt (PDF)
... Translate the Design Model into **VHDL** or Verilog HDL ... to HDL simulation or HDL logic **synthesis**). ... Overlay Operation Sequence onto **Control Structure** Add Hardware ...
www.cse.sc.edu/~jimdavis/Courses/CSCE-611/CSCE611(Fall02)-Lecture11.pdf - View as html - More pages from this site

Results Page:
1 2 3 4 5 6 7 8 9 10 **Next**

WEB RESULTS 21 - 40 out of about 429 (What's this?)

21. DBLP: George K. Papakonstantinou
... Panayotis Tsanakas: Automatic Hardware **Synthesis** of Nested Loops Using UET Grids and **VHDL**. ... George K. Papakonstantinou: A **Control Structure** for a ...
www.informatik.uni-trier.de/~ley/db/indices/a-tree/p/Papakonstantinou:George_K=.html - 23k - Cached

22. A Tool for Mapping aC Description into a Hardware Configuration ... (PDF)
... hardware as a high-level **synthesis** problem ... the intermediate code into correspondent **VHDL** expression ... b) Vertex **control structure** implementation In addition to the ...
matrix.inesc-id.pt/~pff/tfc03/anton-dcis98.pdf - View as html

23. Mixed Abstraction Level Hardware **Synthesis** from SDL for Rapid ... (PDF)
... in cases when a complete **control structure** could be ... and the communication intensive
parts to RT **synthesis**. ... can be described using a single **VHDL** description in ...
www.fzi.de/sim/publications/1999005-paper.pdf - View as html - More pages from this site

24. ACK An Asynchronous Circuit Design Framework (PDF)
... concurrency local datapath resources and partitioned **control structure** CONTROL PARTITION
1 ... Datapath **Synthesis** Generate and synthesize **VHDL** for arithmetic ...
www.cs.utah.edu/acs/publications/sun_slides.pdf - View as html - More pages from this site

25. Design Page (Hardware System Design)
... realised the shortcomings of Verilog and **VHDL** and am ... issue is that the normal **synthesis**
semantics of ... the concept of unbounded loops in the **control structure**. ...
www.cl.cam.ac.uk/users/djg/el.html - 8k - Cached

26. A Full-Custom Self-Timed DSP Processor Implementation (PDF)
... a 16-bit fixed point architecture and data-stationary **control structure**. ... a bootstrap loader were made with **VHDL**, and netlist made with **synthesis** w as ...
www.imec.be/esscirc/papers-97/172.pdf - View as html

27. Overview of **VHDL** (MICROSOFT POWERPOINT)
www.uoguelph.ca/~gardnerw/courses/cis6090/lectures/csp++.ppt - View as html

28. **VHDL** Signal Analysis in VALET (PDF)
... i) control flow of con- current statements; (ii) **control structure** of sequential ... Ieee p1076.6 standard for **vhdl** register transfer level **synthesis**. ...
pages.cpsc.ucalgary.ca/~costi/Articles/PACRIM99.pdf - View as html - More pages from this site

29. VTBYU Project
... The final step of **synthesis** and place-and-route on the ... it will provide as output

as set of synthesizable VHDL models each ... Generation of the **Control Structure**. ...
www.ccm.ece.vt.edu/rtr/vtbyujanus.html - 13k - Cached

30. Visualization and Measurement of Source Code - Dec 97
... and nondisruptive visualization of **control structure** [1, 2 ... of paramount importance for users of **VHDL**. ... interests include high-level **synthesis**, hardware-software ...
www.stsc.hill.af.mil/crosstalk/1997/12/visualization.asp - 25k - Cached

31. A Formal Model for a **VHDL** Subset of Synchronous Circuits (POSTSCRIPT)
... functions of the basic statements according to the **control structure** of the ... formal **synthesis**), but are also essential for **synthesis** and simulation, to ... in **VHDL**. ...
goethe.ira.uka.de/fsynth/publications/postscript/EiKM96.ps - View as html

32. SCHEDULING IN THE PRESENCE OF CONDITIONAL CONSTRUCTS AND ... (PDF)
... CONSTRUCTS AND OPTIMISATION OF **CONTROL STRUCTURE** John Ainscough ... control structures
and **VHDL** processes to ... with a conventional behavioural **synthesis** system, the ...
www.elec.aca.mmu.ac.uk/Pubs/SchedulingInThePresence.pdf

33. ACC_RELEASE_NOTES.txt - Editor (PDF)
... directive on top of the **control structure** for which ... handle binary string for **synthesis** and simulation ... possible to satisfy requirment of **VHDL** simulator – added ...
www.trias-mikro.de/pdfs/Productreleases/AccelFPGA%20Release%20Notes_V1.7.pdf - View as html

34. <html> <head> </head><body><pre><html> <head> </ ...
... of the basic statements according to the **control structure** of the ... **synthesis**), but are also essential for **synthesis** and simulation ... Synchronous designs in **VHDL**. ...
www.ubka.uni-karlsruhe.de/vvv/1996/informatik/80/80.text - 20k - Cached

35. Reconfigurable Smart Components for Next Generation Information ...
... S., and Hendrix, TD (1998) "The **Control Structure** Diagram: An ... CSP to Formally Verify
Compilation of **VHDL** for High Level **Synthesis**," Auburn University Ph.D ...
www.eng.auburn.edu/irsc/RSC_5.htm - 33k - Cached

36. Score:_____Name: ... (MICROSOFT WORD)
... 13. (5 points) Implement the following C **control structure** statement using ... (15 points) Add the instruction, SLTU, to the original MIPS **VHDL synthesis** model. ...
www.ewh.ieee.org/soc/es/Nov1997/01/CSMATRL/2510F1B.DOC - View as html

37. Volume 15 2001
... Tero Tapani (Finland): "A Fuzzy **Control Structure** for A ... Ian A. (Ireland): "Modeling, Simulation and **Synthesis**: From Simulink to **VHDL** Generated Hardware ...
200.75.139.138:1081/ProceedingSCI/volumeXV2001.htm - 45k - Cached

38. VIUF SPRING 93
... model view can be defined by analyzing only the **control structure** information in ... Partitioning
for Multicomponent **Synthesis** from **VHDL** Specifications Authors ...
www.vhdl.org/viuf/papers/VIUF_SPRING_93.html - 38k - Cached

39. **Synthesis** of Low-Power Selectively-Clocked Systems from High- ... (PDF)
... Structures and Microprogramming]: **Control Structure** Performance Analysis ...

imperative
semantics (eg Verilog HDL, **VHDL**). ... power controller **synthesis** exploits the ...
chronos.stanford.edu/users/nanni/publications/archive/2000/TODAES311-321.pdf - View as html

40. Communication (MICROSOFT POWERPOINT)
... **Control structure** is responsible for synchronization eg fork-join. ... Wait statements in **VHDL**. a <=30;. ... denote performance constraints, to be used by **synthesis** tools. ...
www.facweb.iitkgp.ernet.in/~anupam/communication.ppt - View as html

Results Page:
Prev 1 2 3 4 5 6 7 8 9 10 **Next**